

데이터베이스

관계형 데이터베이스

13. 인덱스


create index

(,)

first
coding

- 인덱스를 왜 사용하는 것일까?
 - “빠른 검색을 위해서 인덱스를 사용”
 - 일반적으로 책 맨 뒤에 있는 색인(인덱스, 찾아보기)에서 해당 단어(테이블)를 찾아 그 페이지로 이동.
 - 이렇게 원하는 단어를 쉽게 찾는 방법으로 색인, 인덱스가 사용되는 것처럼 오라클의 인덱스 역시 원하는 데이터를 빨리 찾기 위해서 사용.
- 인덱스란 SQL 명령문의 처리 속도를 향상시키기 위해서 컬럼에 대해서 생성하는 오라클 객체.

01 조회를 빠르게 하는 인덱스

- 인덱스는 장점만 있는 것이 아니다.
 - 오라클에서의 인덱스의 내부 구조는 B* 트리 형식으로 구성.
 - 컬럼에 인덱스를 설정하면 이를 위한 B* 트리도 생성되어야 하기 때문에 인덱스를 생성하기 위한 시간도 필요하고 인덱스를 위한 추가적인 공간이 필요.
 -  인덱스가 생성된 후에 새로운 행을 추가하거나 삭제할 경우 인덱스로 사용된 컬럼 값도 함께 변경되는 경우가 발생.
 - 인덱스로 사용된 컬럼 값이 변경되는 이를 위한 내부 구조(B* 트리) 역시 함께 수정 돼야 함.
 - 이 작업은 오라클 서버에 의해 자동으로 처리되기 때문에 인덱스가 없는 경우 보다 인덱스가 있는 경우에 DML 작업이 훨씬 무거워지게 됨.

()

- 인덱스의 장점
 - 검색 속도가 빨라진다.
 - 시스템에 걸리는 부하를 줄여서 시스템 전체 성능을 향상시킨다.
- 인덱스의 단점
 - 인덱스를 위한 추가적인 공간이 필요하다.
 - 인덱스를 생성하는데 시간이 걸린다.
 - 데이터의 변경 작업(INSERT/UPDATE/DELETE)이 자주 일어날 경우에는 성능이 저하된다.

- 인덱스는 기본 키나 유일 키와 같은 제약 조건을 지정하면 따로 생성하지 않더라도 자동으로 생성.
 - 기본 키나 유일 키는 데이터 무결성을 확인하기 위해서 수시로 데이터를 검색하기 때문에 빠른 조회를 목적으로 오라클에서 내부적으로 해당 컬럼에 인덱스를 자동으로 생성.

- 인덱스가 아닌 컬럼으로 검색하기

```
CREATE TABLE EMP01  
AS  
SELECT * FROM EMP;
```



```
INSERT INTO EMP01 SELECT * FROM EMP01;  
INSERT INTO EMP01 SELECT * FROM EMP01;  
INSERT INTO EMP01 SELECT * FROM EMP01;  
.....
```



```
INSERT INTO EMP01 (EMPNO, ENAME) VALUES(1111, 'KING');
```



```
SELECT DISTINCT EMPNO, ENAME FROM EMP01 WHERE  
ENAME='KING';
```

- CREATE INDEX 명령어로 직접 인덱스를 생성.

```
CREATE INDEX index_name  
ON table_name (column_name);
```

- CREATE INDEX 다음에 인덱스 객체 이름을 지정.
- 어떤 테이블의 어떤 컬럼에 인덱스를 설정할 것인지를 결정 하기 위해 ON 절 다음에 테이블 이름과 컬럼 이름을 기술.

```
CREATE INDEX IDX_EMP01_ENAME ON EMP01(ENAME);
```

- 인덱스 삭제를 위해서 오라클은 DROP INDEX 명령어를 제공 .

```
DROP INDEX index_name;
```

```
DROP INDEX IDX_EMP01_ENAME;
```